A11103 090070

COMPUTER SCIENCE & TECHNOLOGY:

A KEY NOTARIZATION SYSTEM FOR COMPUTER NETWORKS

QC
100
.U57
NO.500-54
1979
c.2

NBS Special Publication 500-54
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities[2] — Radiation Research — Thermodynamics and Molecular Science — Analytical Chemistry — Materials Science.

THE NATIONAL ENGINEERING LABORATORY provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering[2] — Mechanical Engineering and Process Technology[2] — Building Technology — Fire Research — Consumer Product Technology — Field Methods.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

Programming Science and Technology — Computer Systems Engineering.

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Washington, DC 20234.
[2]Some divisions within the center are located at Boulder, CO 80303.

NOV 13 1979

COMPUTER SCIENCE & TECHNOLOGY:

A Key Notarization System for Computer Networks

Miles E. Smid

Operations Engineering Division
Center for Programming Science and Technology
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

Special publication

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Luther H. Hodges, Jr., Under Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

October 1979

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in the series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-54
Nat. Bur. Stand. (U.S.) Spec. Publ. 500-54, 35 pages (Oct. 1979)
CODEN: XNBSAV

Library of Congress Catalog Card Number: 79-600160

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1979

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20234
Stock No. 003-003-02130-0 Price $1.75
(Add 25 percent additional for other than U.S. mailing)

The author would like to express his appreciation to Dr. Dennis Branstad for orginally presenting the key management problem and for his many valuable comments and suggestions.

TABLE OF CONTENTS

A Key Notarization System For Computer Networks

Miles E. Smid

A cryptographic, Key Notarization System is proposed for computer networks to protect personal (nonshared) files, to communicate securely both on and off-line with local and remote users, to protect against key substitution, to authenticate system users, to authenticate data, and to provide a digital signature capability using a nonpublic key encryption algorithm. The system is implemented by the addition of key notarization facilities which give users the capability of exercising a set of commands for key management as well as for data encryption functions. Key notarization facilities perform notarization which, upon encryption, seals a key or password with the identities of the transmitter and intended receiver.

Key words: Cryptography; digital signatures; encryption; identifiers; key management; key notarization.

1. INTRODUCTION

This paper proposes a Key Notarization System (KNS) which may be used in conjunction with a cryptographic device to provide increased data security. In 1977 the National Bureau of Standards published a completely defined encryption algorithm known as the Data Encryption Standard (DES) which became a Federal standard for the protection of unclassified data [2]. Since publication, several companies have produced hardware devices which implement the standard, and there has been an increased awareness that, in certain applications, encryption offers the only effective means of protecting information. The first applications of the encryption of unclassified data appeared in the area of electronic funds transfer, but the passage of the Privacy Act of 1974 (5 USC 522a) and Transmittal Memorandum No. 1 to Office of Management and Budget Circular A-71 placed added responsibilities on Federal data systems for the protection of nonfinancial data as well.

Even before the DES was adopted, it was clear that there was more to cryptographic security than a secure encryption algorithm. Efforts were initiated by NBS to have additional standards, based on the DES, developed. An area which needed to be addressed was secure key management. DES keys are 64-bit binary vectors which are individually selected in order to provide the unknown quantity necessary for security in the encryption algorithm. Key management involves the secure generation, distribution, and storage of cryptographic keys. If the key management is weak, then the most secure cryptoalgorithm will be of little value. In fact, a very strong cryptoalgorithm used in a weak key management system can give a false sense of security.

Previous work on key management systems may be found in Ehrsam, et al [4] and Everton [5]. This paper develops a simple key hierarchy and a set of commands or protocols which in conjunction with a secure random key generator and a strong encryption algorithm may be used to generate and store keys as well as to encrypt and decrypt data. These commands have been devised for computer systems which employ key notarization facilities (KNF's). They are to be tested on the NBS UNIX system but they are not UNIX dependent. It is intended that the system be applicable to many different situations. On-line communications, file encryption, off-line mail, and digital signatures all are to be protected. Key notarization is presented to help provide security while maintaining the required flexibility.

2. REQUIREMENTS

The Key Notarization System (KNS) may be used in computer networks along with key notarization facilities (KNF's) to:

1. Securely communicate between any two users;
2. Securely communicate via encrypted mail (off-line);
3. Protect personal (nonshared) files;
4. Provide a digital signature capability.

Secure communication involves preventing the disclosure of plain text, detecting fraudulent message modification, detecting fraudulent message insertion or deletion, and detecting fraudulent replay of a previously valid message.

The KNS must be consistent with these goals and yet operate at speeds sufficient for normal network communications.

With mail encryption, data is encrypted and then sent via mail or some means which cannot provide an immediate response. The data is stored in the encrypted form until decryption at some later time. In this situation one cannot have an interactive system for exchanging keys because no real-time response is possible. Therefore, protocols must be devised so that the receipt of keys need not be immediately acknowledged.

Once encrypted, personal files can only be decrypted by the original owner. They are encrypted for secure storage rather than secure communication. In this case encryption is used to protect against accidental disclosure, such as spillage, and intentional disclosure, such as scavenging. It is often desirable that the data encrypting key be stored with the cipher for ease of recovery. Of course, the key would be encrypted under another long term key which is kept for the user either in the KNF or in a secure location from which it may be entered into the KNF.

Digital signatures were developed in conjunction with public key systems. (See Diffie and Hellman [3] and Rivest, et al [8].) In such systems the decryption key is not equal to, and cannot be computed from, the encryption key. Encryption keys may be made public while decryption keys are kept secret. A digital signature is decrypted using the secret decryption key and sent to the receiver. The receiver may encrypt, using the public key, and verify the signature, but the signature cannot be forged since only the transmitter knows the secret decryption key. (The cryptoalgorithm must have the property that decryption of the signature followed by encryption equals the original signature.) Popek and Kline [7] showed that nonpublic key algorithms can also be used for digital signatures in conjunction with a "Network Registry". In the KNS, a different method is proposed for implementing digital signatures with the DES nonpublic key algorithm.

3. SYSTEM DESIGN

3.1 THE NETWORK

The KNS is designed for computer networks which consist of host computers, user terminals, and key notarization facilities. Figure 1 shows a four host network. The host controls the normal operation and communication of the terminals. Terminals have the capability of communicating with the host, with other local terminals through the host, and with terminals of other hosts via communication channels called interchanges. Each terminal will be able to use the host KNF by means of user commands. All commands will be implemented in the KNF, and every KNF will have the capacity to generate keys for distribution to other hosts or facility users.

Interchanges may be electronic communications lines, microwave links, courier routes, etc., or combinations of more than one medium. In Figure 1 only host 3 shares an interchange with host 4. If host 1 shares a common interchange key with host 4, then host 1 may communicate with host 4 through host 3 without intermediate decryption and reencryption. Host 3 would merely act as a switch. This is known as end-to-end encryption. If host 1 does not share a common key with host 4 but does share a key with host 3, and if host 3 shares a key with host 4, then host 1 may communicate with host 4 via host 3. The cipher would have to be decrypted at host 3 and reencrypted in the key shared between host 3 and host 4. Care must be taken to insure that the communications are not compromised when unencrypted. This method of encrypted communications is called link encryption.

The lines between the KNF and its host and the lines between each terminal and its host must be protected. They could be physically secured or they could be secured by the addition of cryptographic devices on each end of the links. When a user is editing a file in the host, it is in plain text form, and the host will have to protect the data from other users. Once the user has finished editing, he may command the KNF to encrypt the data and store the resulting cipher in unprotected memory or send it to a remote user over an interchange.

3.2 THE HOST

We will assume that the host computer has two types of memory: that which is not accessible to any user, called system memory, and that which is accessible to users, called user memory. User i's memory is core, disk, etc., where user i is permitted to store and recall data. Most computers have a means of protecting system memory from users, and some computers protect one user from another to a certain degree. We will rely on these protective features to the extent that the user should not be able to subvert the operation of the computer. For example, the system must be able to correctly maintain the identity of the user once he has been authenticated and given permission to execute the commands. The system must also prevent one user from taking on the identity of another user and thereby obtaining access to his unencrypted data. In other words, encryption by itself does not solve the computer security problem. However, if properly used in a system with the necessary protective features, it can provide protection to stored and communicated data.

The encrypted keys of user i are stored in user i's memory, and encrypted passwords to which no user needs access will be stored in system memory. Nevertheless, we will assume that any user could gain read and write access to every encrypted password stored in system memory. Each user is expected to manage the encrypted keys which belong to him, but he will not know any clear keys. Yet, key encryption is not sufficient. A method is required to protect against key substitution and to insure that each user correctly identifies the user with whom he is communicating.

3.3 THE KEY NOTARIZATION FACILITY (KNF)

The KNF contains a DES encryption device. It will have a control microprocessor and memory to implement commands and data transfers. The KNF must also store the unencrypted interchange keys and the states of active users. An active state consists of a user identifier along with an initialization vector and an unencrypted data key for both transmitting and receiving data. A user is active as soon as his identifier is loaded into active user memory in the KNF. He may then proceed to load the rest of his state.

The KNF contains a key generator which is capable of generating unpredictable keys. At any time a user should be able to predict the next key to be generated with only a 1/(2**56) probability of success where 2**56 is two raised to the 56'th power. One possible key generator is proposed in the Appendix. Once the 56-bit keys are generated the

proper parity is determined and the entire 64-bit key is encrypted before it is returned to the host. Thus, no clear keys are known outside the KNF. The key generator is also used to generate 64-bit initialization vectors which initialize the DES cryptoalgorithm. Since the KNF contains clear keys, the encryption algorithm, the commands program, and the key generator, it must be physically protected.

Cryptographic facilities containing a single master key are used in Ehrsam, et al [4] to perform encryption and execute key management commands. Our key notarization facilities hold several keys and the key generator. They employ a different key hierarchy, a different set of commands, and are the enforcers of key notarization.

3.4 DISTRIBUTED VERSUS CENTRALIZED KEY GENERATION

Branstad [1] describes how network security centers (NSC) may be used for key distribution. Upon request, the NSC generates a key for use by each of the parties in a conversation. One copy is encrypted under a key shared between the NSC and the first party and another copy is encrypted under a key shared between the NSC and the second party. The encrypted forms of the key are then sent to the appropriate receivers.

The KNS uses distributed rather than centralized key generation as employed by an NSC. In order to provide for off-line encrypted mail, the KNS gives each host the capability of key generation in its own KNF. Thus, two hosts do not even have to be electronically connected in order to communicate securely. The KNS requires fewer protocols because parties do not have to send a remote key generation request and they do not have to respond to the receipt of a key. Fewer protocols mean fewer ways an enemy can attempt to trick or confuse the communicating parties by altering or playing back the protocol messages. (See Needham and Schroeder [6].)

If a KNF is compromised, only communications involving the compromised facility are compromised. If a NSC is compromised, and there is only one NSC for the network, then the whole network is compromised. Finally, with a local key generator, one can encrypt personal (nonshared) files without having to depend on a remote site. The KNS approach has the disadvantage that the key generation capability and the KNF physical security has to be replicated at each host.

4. IDENTIFIERS AND KEY NOTARIZATION

A special feature of the KNS is the support of key notarization. This feature increases security, permits a simple system design, and provides a means of implementing signatures with a nonpublic key system. Identifiers are nonsecret binary vectors of up to 28 bits which uniquely identify each user in the network. When a user first attempts to call the KNF he must submit his identifier along with the correct password to establish an active state in the KNF. Both the host and the KNF employ identifiers to "recognize" the users.

Key notarization is similar to the actions of a notary public who first requires his customer to identify himself via a driver's license, etc., before he seals (notarizes) the customer's signature on a document with his notary stamp. In addition to the notary's function of authenticating the creator of a message, the KNS authenticates the message itself and the person requesting decryption. Key notarization is similar to having a notary public on each end of a secure communication channel.

Let i and j be identifiers and K be a DES key. Then (i || j) represents the concatenation of i and j . K, a 64 bit key, consists of eight btyes, each with seven information bits and a parity bit. K XOR (i || j) is a special function defined as follows. The leftmost seven information bits of K are exclusive or'ed with the leftmost seven bits of i. The eighth bit, a parity bit, is then appended so that the modulo 2 sum of all eight bits is odd. Then the next seven information bits of K are exclusive or'ed with the next seven bits of i and the correct parity bit is appended. This continues until the last seven information bits of K have been exclusive or'ed with the last seven bits of j and the final parity bit has been set. Therefore, K XOR (i || j) is a valid DES key with 56 information bits and eight parity bits.

All passwords and data keys are encrypted under K XOR (i || j) for some K and some i, j pair. In the case of passwords i = j. This adds to security because one user cannot substitute his password or keys for those of another user and be able to authenticate or decrypt as that user. This will be explained in detail in section 7, PASSWORD AND KEY STORAGE. The security is also increased because both parties in a conversation must know the other's correct identity to communicate. Since the KNF only needs to retain keys for each interchange, instead of each user, the network design is simplified; and since only one user can encrypt with a given data key and only one user can decrypt with a

given data key, a signature system may be devised similar to those used with public key encryption systems. This will be discussed further in section 10, DIGITAL SIGNATURES.

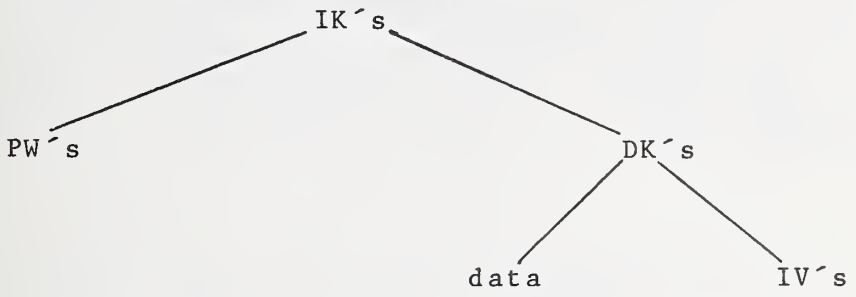
When key notarization is used, keys and passwords are sealed, upon encryption by the KNF, with the identifiers of the transmitter, or key generator, and the receiver. To generate a notarized key the transmitter must identify himself to the KNF and provide proof of his identity by supplying his correct password. We call this user authentication. He must also identify the intended receiver of the key. Once encrypted, the correct key cannot be decrypted unless the correct identifier pair is again provided. To decrypt the key the receiver identifies himself and provides password proof of his identity. The receiver must also supply the identity of the transmitter which may have been sent unencrypted. If the identification information is not the same as that provided by the transmitter to his KNF, then the decrypted key will not equal the original key and no information can be correctly decrypted. Thus, the receiver must know the correct transmitter and be the intended receiver.

5. USER AUTHENTICATION

Each user will have a password which is used to authenticate the user and permit him to invoke user commands. The plain password is passed through an encryption function, involving the user's identifier, and the result is compared with a stored value before the user is activated. Therefore, a user cannot exercise any other command until his identity has been authenticated. The password of each user is stored in system memory encrypted under the facility interchange key (See section 6, KEY HIERARCHY.) combined with the user's identifier. Since it is assumed that the host can maintain the correct identity of a user once he has been authenticated, the user need not resubmit his password for each key he generates while he is active. His authenticated identifier which has been loaded into active user memory will automatically be used as his identifier.

6. KEY HIERARCHY

Two distinct types of keys are used to form the key hierarchy, interchange keys (IK's) and data keys (DK's). Interchange keys encrypt passwords (PW's) and data keys, while data keys encrypt both data and initialization vectors (IV's). The key hierarchy is shown below.



6.1 INTERCHANGE KEYS (IK's)

Interchange keys are used for the exchange of keys between users. One interchange key, called the facility interchange key, is used for communication within a facility and the encryption of facility user passwords. Other interchange keys may be available for the exchange of data keys between facilities or for special subgroups of a facility. IK's are generated outside the network and are entered, unencrypted, directly into the KNF. This permits two facilities to enter the same IK. One IK can be used to connect all the users of two hosts since a user may not decrypt a data key shared by two other users. This is because the identifiers of the two parties are involved in the encryption of the shared key. Therefore, the number of keys which need to be stored in the KNF is reduced.

6.2 DATA KEYS (DK´s)

Data keys are used to encrypt data belonging to one particular user or data shared between two users. DK´s are generated by the key generator and are immediately encrypted under an IK XOR´ed with the proper identifier pair. The identifier of the user requesting the key, who is also the transmitter, is always the left identifier and the identifier of the intended receiver is the right identifier in the identifier pair. When encrypted, DK´s may be sent, kept in unprotected memory, etc.. Initialization vectors are employed by the DES algorithm in the cipher block chaining (CBC), cipher feedback (CFB), and data authentication (DAUT) modes of operation. All IV´s are encrypted, before they leave the KNF, under the data key which enciphers the corresponding data.

7. PASSWORD AND KEY STORAGE

Figure 2 shows how keys appear in KNF memory at host 1, in host 1 system memory, and in the memory of user i at host 1.

7.1 KNF KEYS

KNF memory contains both current and old interchange keys and active states of a limited number of users. When the interchange keys are changed, the old interchange keys are securely stored outside of the KNF along with their effective date. With the addition of another command one could encrypt the IK´s in the facility master key to reduce the number of clear keys needing protection. The current IK´s become the old interchange keys and the new interchange keys become the current IK´s. After such a change, the passwords are reencrypted under the current (new) facility interchange key, and the users are told to reencrypt their data keys.

7.2 PASSWORDS

System memory contains the encrypted passwords for every user. Let E[X](Y) indicate the encryption of Y under X in the electronic codebook (ECB) mode of operation. Thus, E[IK1 XOR (i || i)](PWi) denotes the encryption of PWi under IK1 XOR´ed with user i´s identifier pair, (i || i). IK1 is used because the encrypted passwords are from the system

memory of host 1 and IK1 is the facility interchange key for host 1.

The password is encrypted under IK1 XOR´ed with the appropriate identifier pair to protect against substitution. If identifiers were not used, system memory might appear as follows:

i. E[IK1](PWi)

j. E[IK1](PWj)

.

.

If user j could gain access to system memory, he might alter it as follows:

i. E[IK1](PWj)

j. E[IK1](PWj)

.

.

User j could then authenticate as user i by submitting his own password while claiming to be user i. If identifiers are used as in figure 2, then E[IK1 XOR (i || i)](PWj) would be calculated upon authentication and it would not compare with E[IK1 XOR (j || j)](PWj) which was substituted as user i´s encrypted password.

7.3 USER KEYS

User i´s memory contains personal and shared data keys. Personal data keys are encrypted under the facility interchange key XOR´ed with the user´s identifier pair. Personal keys may be used to encrypt files and other private data, but cannot be shared. User i´s memory also contains shared data keys encrypted under interchange keys XOR´ed with the concatenation, (||), of user i´s identifier and another user´s identifier. (i || j) uniquely identifies the communication parties. If (i || j) were not used, another user could substitute his own data key encrypted under the

interchange key and then be able to decrypt any subsequent cipher. Similarly, when user j receives E[IKp XOR (i || j)](DKij), he must know that he is communicating with i, over interchange p, to correctly decipher DKij. Thus, the transmitter is prevented from posing as someone else. Since several users may all use the same IKp to communicate, this protection is critical.

It should be noted that it is the system's responsibility to enforce any restrictions on the use of interchanges. For example, if user i is not allowed to use IKp then the system must enforce this arbitrary restriction by not loading IKp for user i. However user i should not be able to subvert the restriction by key substitution.

One could argue that substitution protection is not needed for system memory because if the system cannot protect system memory, it probably cannot prevent users from changing identity, from invoking system commands, or other security threats. This may be true but encryption should not add additional possibilities for attacks. In user memory the substitution threat is very real because many systems cannot protect one user's memory from another user, and even if they could, the encrypted keys will not be protected. Encrypted data keys may be stored with cipher on unprotected tapes and disks, and they may even be sent out over unprotected communications channels.

8. DEFINITION OF TERMS

When defining our commands, the terms, initialize, reserve, load, store, generate, encrypt, decrypt, and reencrypt will be used. These terms should be defined so that the meaning of the commands is clear. The terms actually represent functions which operate on keys or passwords.

initialize: Sets a password to a starting value that should be changed by invoking another command.

reserve: Activates a user by loading his identifier into the KNF.

load: Takes an encrypted key or encrypted IV from the user, decrypts it, and puts it into the active user memory in the KNF.

store: Places an encrypted password in system memory. Operates on PW.

generate: Calls the KNF random key generator which generates 56 unpredictable, random bits that are combined with eight parity bits, as required by the DES. The result is encrypted under an interchange key XOR´ed with the appropriate identifier pair. IV generation provides a full 64 random bits before encryption. Operates on DK´s and IV´s.

encrypt: Encrypts a DK or PW under an IK XOR´ed with the appropriate identifier pair. Uses the ECB mode of encryption when operating on keys. "Encrypt" also refers to enciphering data in one of the approved DES modes.

decrypt: Decrypts an encrypted DK or PW. "Decrypt" also refers to deciphering data in one of the approved DES modes.

reencrypt: Decrypts an encrypted DK or PW and then encrypts it under a new IK XOR´ed with the appropriate identifier pair in order to avoid the reencryption of data and the reinitialization of passwords when IK´s are changed.

9. COMMANDS

This section describes the commands or protocols which need to be implemented in the KNF for key management and data encryption purposes. Besides encryption, decryption, and authentication, they are used to generate keys which are given to the user and to provide for the supersession of the keys which are controlled by the system. The commands are invoked by a command name followed by a parameter address list of passed and returned values. The user´s identifier is shown as a parameter only when it must be supplied by the user of the command. For some commands the system automatically supplies the KNF with the user´s identifier. Interchange keys must be loaded into the KNF before commands are executed.

9.1 INITIALIZE PASSWORD (IPW)

```
IPW: {pw}
pw = password
```

This command is used when a user is first put on the system. The password is encrypted and stored in host system memory. The original password is known to the user and the security officer. The user submits the original password when he first authenticates himself to the KNF, then he immediately changes his password to a secret value, known only to himself, by using the change password command, CPW. Only the security officer who is responsible for putting new users on the system should be capable of initializing the password.

9.2 REENCRYPT PASSWORDS (RPW)

```
RPW: { }
```

The security officer executes this command after the interchange keys have been changed. Each encrypted password stored in system memory is decrypted using the old facility interchange key and encrypted using the new facility interchange key. The result is then stored back in system memory. This permits a user to authenticate even though the interchange keys have been changed. After he is authenticated and active, it will be the user's responsibility to reencrypt his data keys before using them for encryption, decryption, or data authentication.

9.3 RESERVE ACTIVE STATE (RAS)

```
RAS: {ui, pw, ss, ua}
ui = user identifier
pw = password
ss (system status)  = y if active memory is available
                    = n otherwise
ua (user            = 0 if ss = n
    authenticator)  = y if ss = y and PW authenticates
                    = n if ss = y and no authentication
```

This command activates the user by loading the user's identifier into the KNF. Active user memory must be available and the user must authenticate before the identifier is loaded. No other commands may be executed by the user until he has successfully executed RAS. The authentication is for use of the KNF and is independent of the authentication for

use of the system. Once authentication is complete, the system must insure that other users cannot execute commands in place of an authenticated user.

9.4 LOGOUT ACTIVE USER (LAU)

LAU: {ui}
ui = user identifier

This command may be used by the user when he has finished using the KNF. In this case ui is optional. The command removes the user identifier from the active user list maintained in the KNF. All active DK's and IV's belonging to the specified user are lost. The host may also keep a list of active users and the time of the last command executed for each one. If a user has not executed a command after a reasonable time period, then the host may use LAU to log out the user. The user may still be logged on the system but he will have to repeat the RAS command to use the KNF. The system may also periodically decide to challenge a user by requiring him to reauthenticate. Whenever the user logs off the system, the LAU command should automatically be executed.

9.5 CHANGE PASSWORD (CPW)

CPW: {op, np}
op = old password
np = new password

This command is used to change passwords. The old password is authenticated before any change is made. The user identifier must be loaded into active user memory, otherwise an error message is returned.

9.6 GENERATE DATA KEY (GDK)

GDK: {in, sp, ed}
in = interchange name
sp = identifier of sharing party
ed = returned encrypted data key

ex. (command executed by user i)
in = p
sp = j
ed = E[IKp XOR (i || j)](DKij)

This command is used to generate new keys. The identifier of the user invoking the command, user i in the example, is always the leftmost value in the concatenation of the sending and receiving identifiers. If the two identifiers are equal, then the key is personal and cannot be shared. This command may not be executed unless the user is active. Otherwise an error message is returned.

9.7 ENCRYPT DATA KEY (EDK)

```
EDK: {ui, dk, ed}
ui = user identifier
dk = data key
ed = returned encrypted data key

ex.
ui = i
dk = DK
ed = E[IK XOR (i || i)](DK)
IK = facility interchange key
```

This command is not used in the normal functioning of the system. It need only be used for communication with someone outside of the system who doesn't have the same key generation and encryption capability or for generating cipher encrypted under a particular key. Since this command violates the security criterion that no clear key be permitted outside of the KNF, it is recommended that only the security officer be allowed to execute it. It may be best not to implement this command at all.

9.8 LOAD DATA KEY (LDK)

```
LDK: {kf, in, sp, ed}
kf (key function) = t if key is for transmitted data
                  = r if key is for received data
                  = s if key is for personal use only
in                = interchange name
sp                = identifier of sharing party
ed                = encrypted data key

ex. (command executed by user i)
kf = t
in = p
sp = j
ed = E[IKp XOR (i || j)](DKij)

ex. (command executed by user i)
kf = r
```

```
in = p
sp = j
ed = E[IKp XOR (j || i)](DKji)

ex. (command executed by user i)
kf = s
in = f (facility interchange identifier)
sp = i
ed = E[IKf XOR (i || i)](DKii)
```

This command loads a data key, either shared or personal, into the user's active state in the KNF. The key is stored at the transmit key address if kf = t, and at the receive key address if kf = r. If user i executed the command, then kf = s if and only if sp = i. Otherwise an error message will be returned. When kf = s and sp = i the data key will be loaded into both the transmit and receive locations. The user must be active before this command can be executed.

9.9 GENERATE INITIALIZATION VECTOR (GIV)

```
GIV: {ei}
ei = returned encrypted initialization vector

ex.
ei = E[DK](IV)
```

This command is used to generate new initialization vectors. The KNF key generator generates 64 bits, (56 random and 8 parity), and then encrypts them under the data key which must be previously loaded at the transmit address in active user memory. The encrypted IV is returned to the user. The data key may be either personal or shared.

9.10 LOAD INITIALIZATION VECTOR (LIV)

```
LIV: {kf, ei}
kf = t if IV is for transmitted data
   = r if IV is for received data
   = s if IV is for personal data
ei = encrypted initialization vector

ex.
kf = t
ei = E[DK](IV)
```

If kf = t then the data key at the transmit address is used to decrypt the encrypted IV. The IV is then stored at the transmit IV address. If kf = r then the data key at the receive address is used to decrypt the encrypted IV, and the IV is stored at the receive IV address. When kf = s, the transmit data key is used to decrypt and the IV is placed in both the transmit and receive IV locations.

9.11 ENCRYPT INITIALIZATION VECTOR (EIV)

```
EIV: {iv, ei}
iv = initialization vector
ei = returned encrypted IV

ex.
iv = IV
ei = E[DK](IV)
```

This command is not necessary because one can always use the GIV command to obtain IV's. However, it may be used with the EDK command for communications outside of the system. Since, in the KNS, no unencrypted IV's are to be known by users, it is recommended that this command be restricted solely to the security officer or omitted completely. The IV is encrypted under the DK previously loaded at the transmit key address.

9.12 REENCRYPT DATA KEY (RDK)

```
RDK: {kf, in, sp, ok, rk}
kf = t if data key is for transmitted data
     r if data key is for received data
     s if data key is for personal data
in = interchange name
sp = identifier of shared party
ok = old encrypted data key
rk = returned reencrypted data key

ex. (user j reencrypting a key sent to him by user i)
kf = r
in = p
sp = i
ok = E[IKp' XOR (i || j)](DKij)
rk = E[IKp XOR (i || j)](DKij)
IKp' = old interchange key
IKp  = new interchange key
```

This command is used when interchange keys are changed. It

reencrypts data keys under the new interchange key so that the data protected by the key does not have to be reencrypted. The user must be active. Also, kf = s if and only if sp = i and user i invoked the command.

9.13 ELECTRONIC CODEBOOK (ECBE AND ECBD)

```
ECBE: {pt, ct}
ECBD: {pt, ct}
pt = plain text (eight bytes)
ct = cipher text (eight bytes)
```

These commands are not required in the normal operation of the system. They are provided to accommodate future modes of DES encryption which, as yet, have not been considered or approved. ECBE encrypts eight bytes of plain text at pt and stores the result in ct. ECBD decrypts eight bytes of cipher at ct and stores the result at pt. Encryption uses the transmit DK while decryption uses the receive DK. A data key must be previously loaded into the appropriate active state.

9.14 DATA AUTHENTICATION (DAUT)

```
DAUT: {kf, da, nb, av, md}
kf = t if data is transmitted
   = r if data is received
   = s if data is personal
da = data
nb = number of bytes of data
av = returned authentication value (eight bytes)
md = CBC for CBC mode
   = CFB for CFB mode
```

This command uses DES in the authentication mode to calculate an eight-byte authentication value on nb bytes of data at da. If kf = t or s then the data key and IV which have been previously loaded into transmit active storage will be used. If kf = r the key and IV in receive key active storage will be used. The value of md indicates which of two DES encryption modes are desired.

9.15 CIPHER BLOCK CHAINING (CBCE AND CBCD)

```
CBCE: {pt, ct, nb}
CBCD: {pt, ct, nb}
pt = plain text
ct = cipher text
nb = number of bytes
```

For encryption, CBCE, nb bytes of data starting at pt are encrypted in the CBC mode and the cipher is returned starting at ct. For decryption, nb bytes of data at ct are decrypted and returned to pt. If nb is not a multiple of eight, then the CBC mode is used until b < 8 bytes remain. The final b bytes are encrypted by exclusive or´ing them with the first b bytes of the next DES output block. DK and IV must be in the active user memory otherwise an error message is returned. Encryption uses the transmit IV and DK while decryption uses the receive IV and DK.

9.16 CIPHER FEEDBACK (CFBE AND CFBD)

```
CFBE: {pt, ct, nb}
CFBD: {pt, ct, nb}
pt = plain text
ct = cipher text
nb = number of bytes
```

As described for the CBC commands, nb bytes are either encrypted or decrypted. Encryption uses the transmit IV and DK while decryption uses the receive values. If the required IV and DK values have not been loaded an error message will be returned.

10. DIGITAL SIGNATURES

10.1 RATIONALE

Recall that digital signatures are possible with public key algorithms because one cannot decrypt another person´s data even though anyone with the public key can encrypt data intended for that person. This is because the decrypt key is not shared. Since the KNF combines identifiers with interchange keys for protection against substitution and

employs separate encryption and decryption key storage, one cannot encrypt data in a key that was generated by another user. Therefore, signatures are possible. Suppose user i generates a key using the GDK command and sends it to user j. The encrypted data key would be of the form:

$$ED = E[IKp \text{ XOR } (i \,||\, j)](DKij)$$

where IKp is the interchange key for interchange p and DKij indicates a data key generated by i for transmission to j. Whenever i generates a key his identifier is always leftmost in the identifier pair used in the encryption of the key. The only way user j can load DKij is by loading it as a receive key. Separate transmit and receive key registers are required. If j tries to load DKij as a transmission key for the encryption of data going to i, the cryptomodule will use (j || i) instead of (i || j) when decrypting ED. If j tries to load the key as a personal key, then (j || j) will be used. (See the LDK command in section 9, COMMANDS.) When DKij is loaded as a receive key, only the decryption commands have access to it.

10.2 EXAMPLE

Suppose user i generates ED as before. He may then use the EIV command to generate an encrypted IV of the form:

$$EI = E[DKij](IV).$$

Next, he may encrypt a signature, S, under DKij and send ED, EI, and S to j. User j may load IV and DKij in the active receive state by the LIV and LDK commands, and decrypt the encrypted signature to recover S. There is no way that j can alter S to a particular S´ and encrypt it under DKij because there is no way for j to get DKij into the transmit data key active storage.

If user j generates his own encrypted data key, it will be of the form: E[IKp XOR (j || i)](DKji). He may encrypt a signature S´ under DKji but he cannot claim that it came from i because he could be challenged to decrypt the encrypted signature. To do so j would have to load DKji by submitting E[IKp XOR (j || i)] (DKji) to the LDK command with kf = r. The cryptomodule would not load the correct DKji because it would use (i || j) instead of (j || i) as the identifier pair. Thus, the signature would be garbled. Of course, user j may send a signature S´ to user i

encrypted under the data key, DKji, in a similar manner as described above. (See the illustration below.)

Separate Transmit and Receive Key Storage

User i				User j		
transmit:	IV1	DKij	S---------->	receive:	IV1	DKij
receive:	IV2	DKji	<----------S	transmit:	IV2	DKji

Any message may be regarded as a signature. No additional keys or commands are required. All user j needs to do is keep E[IKp XOR (i || j)](DKij), E[DKij](IV), and the encrypted signature in order to be able to prove that S was sent to him from i. User j may also wish to keep S as well.

10.3 THE AUTHENTICATION VALUE AS A SIGNATURE

The signature, S, may be an entire plain text message, but it may be undesirable to store the cipher text of long messages. In such cases one may use the DAUT command to calculate an authentication value which is a cryptographic function of every bit of data. This value could then be used as the signature. Signatures should be large enough to provide adequate security. At least 64 bits are recommended. S must be encrypted as in the previous example. Otherwise, the receiver could modify the message and calculate the correct signature for the new message using the DAUT command with the correct key in the receive active memory. This is because, unlike encryption and decryption, the DAUT command with kf = t gives the same output as when kf = r as long as the same key is used in both transmit and receive active memory.

If one is not concerned with proving that the receiver did not modify the incomming message, then the authentication value need not be encrypted. Suppose that it is only necessary that the receiver knows the correct transmitter of the message and that it has not been altered. The transmitter, user i, may generate E[IKp XOR (i || j)](DKij) and E[DKij](IV), and load DKij and IV into transmit active memory. He may then use the DAUT command with kf = t to generate an authentication value, AV. User i may then send the following to j:

clear message, E[IKp XOR (i || j)](DKij), E[DKij](IV), AV.

User j may authenticate the message by loading DKij and IV into active receive state and then using DAUT with kf = r to calculate AV. If it matches, then the message must have come from i. If user k sent the message, the encrypted data key would have the form: E[IKp XOR (k || j)](DKkj), and the authentication value would be, AV'. If j thought that it was from i, then when he executes LDK, (i || j) instead of (k || j) would be used to decrypt the data key. Therefore, the wrong data key would be loaded.

10.4 NONPUBLIC KEY VERSUS PUBLIC KEY SIGNATURES

A digital signature capability may be implemented in the KNS because the receiver of an encrypted data key can only load the key into his receive active memory and can, therefore, only decrypt with it. We have assumed that the KNF of each host is physically secured from all users and that shared keys are securely distributed. One must guard against both disclosure and substitution of keys. If one could gain knowledge of the shared key, he could forge all signatures sent between both facilities. Of course, all keys encrypted under the shared key would also be compromised. Thus, the common key must be secured at the transmit and receive KNF's. With public key algorithms, the secret key requires protection against disclosure and substitution while the public key must be protected from substitution. If a bogus key is substituted for the transmitter's public key, then false signatures can be sent to the receiver.

11. ILLUSTRATIVE EXAMPLE

11.1 INITIALIZATION

Suppose cryptography were to be added to a computer network. First, each host would have to be provided with a KNF and the necessary interface. Then interchange keys would have to be generated and distributed. Once the interchange keys are loaded directly into the cryptofacilities and the authorized users are assigned unique identifiers and passwords, the security officer at each facility can initialize the passwords of the authorized users by using the IPW command.

11.2 THE TRANSMITTER

A user may then authenticate and become active by using the RAS command. He could change his password to a secret value known only to himself by the CPW command. Next he may want to generate data keys using GDK. Suppose he is on host 1, then GDK: {1, i, ed} generates a personal data key and GDK: {1, j, ed} generates a shared data key for use with user j at host 1. GDK: {5, k, ed} generates a shared data key for use with user k over interchange 5. Interchange 5 may be the interchange between host 1 and host 5.

When he has an encrypted data key, say E[IKp XOR (i || j)](DKij), user i can load the key using LDK. LDK with kf = t, in = p, sp = j, and ed = E[IKp XOR (i || j)](DKij) loads DKij into the transmit active key storage. The user must keep track of the fact that kf = t and in = p from the time the key is generated to when the key is loaded. If the key is stored for future use, then the values of kf and in required by the LDK command should also be stored. User i may then generate an IV using GIV and load the IV into the transmit active IV storage. After he sends the encrypted DKij and IV to j, he is ready to encrypt data intended for user j. Of course if he is on-line with user j, he must establish contact with j, identify himself, and send him the encrypted DKij and IV. If he is on-line he should require an appropriate response from j to insure that he is being received. User i may encrypt in either the CBC or CFB modes. He should include a message number, the date, and the time in his plain text so that old valid messages from i to j cannot be played back to j. He may also use DAUT to calculate a digital signature which is encrypted before transmission.

User i may use his personal encrypted key, E[IK1 XOR (i || i)](DKii) to encrypt a personal file and then store the encrypted key with the cipher or in a personal key file. Finally, he can log out of active status using the LAU command. If not, the system should automatically log him out after a specified time period or when he logs off the system, whichever comes first.

11.3 THE RECEIVER

Once user j is active, and has received the encrypted DKij, IV, and data, he may use LDK and LIV to load the receive active storage. He can then decrypt and check the signature to insure that it is correct. Note that the same data key may be used for both encryption and digital signatures. If j wishes he can generate a DKji to communicate securely to i, but communications from j to i will not be

encrypted with the same data key as communications from i to j.

11.4 KEY SUPERSESSION

Interchange keys are generated in an unpredictable manner in a highly protected environment outside of the network. At key change time, the current IK's are stored as old IK's and new IK's are entered as current IK's. The security officer uses RPW to reencrypt each user's encrypted password. The system tells each user when he becomes active to use the RDK command to reencrypt his data keys. When keys are changed, the old keys no longer stored in the KNF should be securely stored along with their effective dates. These keys may be needed to decrypt old files or to validate old signatures whose data keys were not reencrypted.

12. SUMMARY

The Key Notarization System can provide secure authentication and encryption with limited protocol requirements in a variety of network configurations. Host operating systems must protect plain text and maintain user identity once authentication is complete, but the host need not protect keys from either disclosure or substitution. A set of KNF commands is defined for key management functions as well as for the approved DES modes of operation. The secure distribution of data keys is attained by encryption and the use of identifiers for key notarization. The system features on-line and off-line applications, local key generation, and a digital signature capability.

APPENDIX

The key and initialization vector generator described here has not been analyzed either for its quality as a pseudorandom number generator or for its security. It is merely presented as an example which may be altered or replaced at a later date.

E[X](Y) represents the DES encryption of Y under key, X, in the ECB mode. Let FIK be the facility interchange key, and let V be a seed value. V may be initially set to any number. Let DT be the date-time word that is passed to the KNF from the host on each key or initialization vector generation command. A 64-bit vector, R is generated as follows:

$$I = E[FIK](DT)$$

$$R = E[FIK](I \text{ XOR } V)$$

and a new V is given by $V = E[FIK](R \text{ XOR } I)$.

If the GDK command made the call, then a data key is created by resetting every eighth bit of R so that the modulo 2 sum of the bits of each 8-bit byte is odd. If the GIV command made the call, R is used directly as the 64-bit initialization vector.

REFERENCES

1. Branstad, Dennis K., Encryption Protection in Computer Data Communications, IEEE Fourth Data Communications Symposium, 7-9 Oct. 1975.

2. Data Encryption Standard, National Bureau of Standards (U.S.), Federal Information Processing Standards Publication (FIPS PUB) 46, National Technical Information Service, Springfield, VA (1977).

3. Diffie, W. and Hellman, M., New Directions in Cryptography, IEEE Trans. Inform. Theory, vol. IT-22, Nov. 1976.

4. Ehrsam, W. F., Matyas, S. M., Meyer, C. H. and Tuchman, W. L., A Cryptographic Key Management Scheme for Implementing the Data Encryption Standard, IBM Systems Journal, vol. 17 no. 2, 1978.

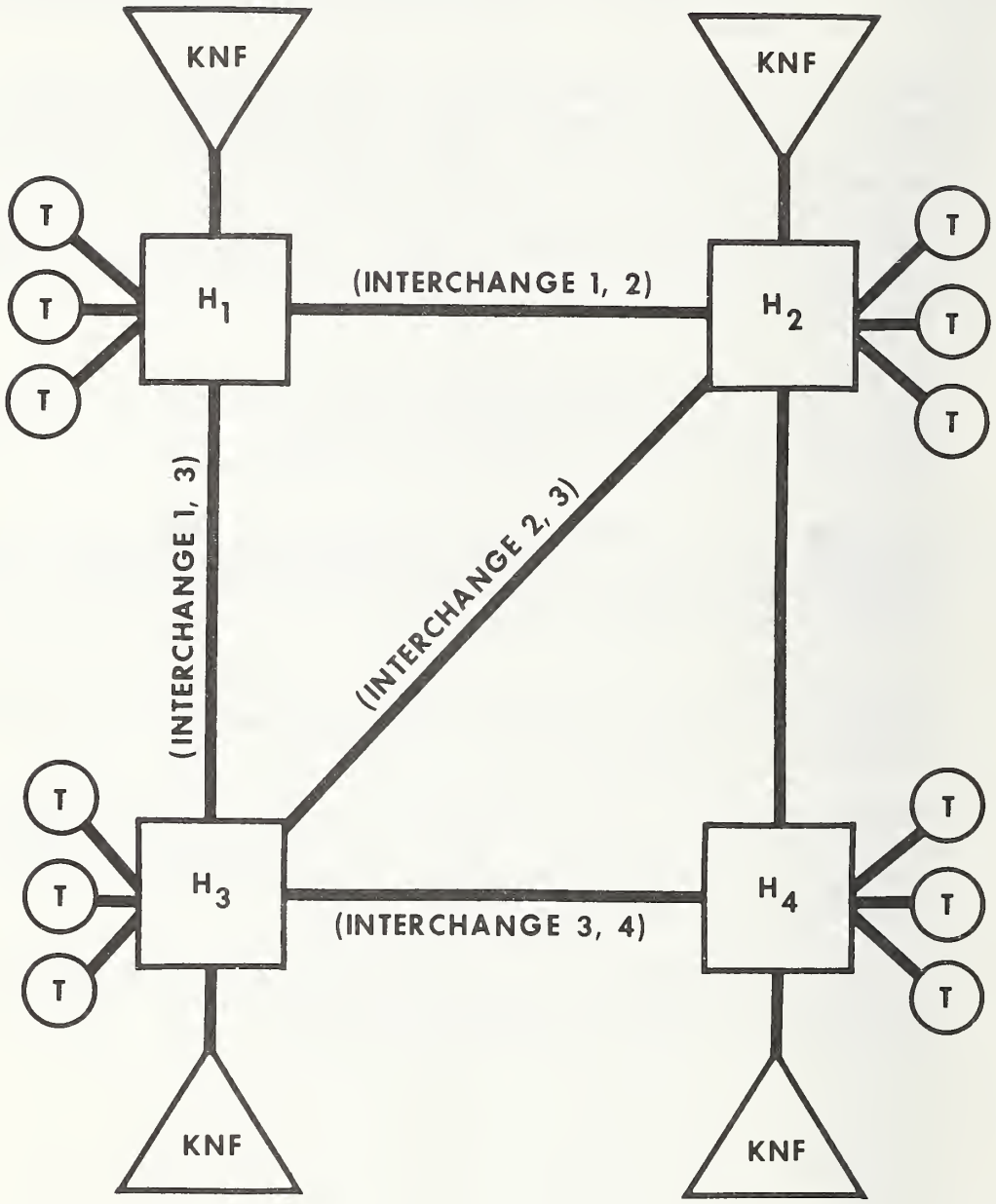
5. Everton, J., A Hierarchical Basis for Encryption Key Management in a Computer Communications Network, Trends and Applications: 1978 Proc., IEEE Computer Society, May 1978.

6. Needham, Roger M. and Schroeder, Michael D., Using Encryption for Authentication in Large Networks of Computers, Communications of the ACM, Dec. 1978.

7. Popek, Gerald J. and Kline, Charles S., Encryption Protocols, Public Key Algorithms and Digital Signatures in Computer Networks, Foundations of Secure Computation, Academic Press, 1978.

8. Rivest, R., Shamir, A. and Adleman, L., A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Commun. ACM, Feb. 1978.

FIGURE 1:
A FOUR HOST NETWORK



KNF = KEY NOTARIZATION FACILITY

H_i = HOST i

T = TERMINAL

Figure 2:
Password and Key Storage

KNF MEMORY AT HOST 1

Current	Old
IK1	IK1´
IK2	IK2´
IK3	IK3´
.	.
.	.
.	.

	Transmit		Receive	
i.	IVt	DKt	IVr	DKr

(for a limited number of active users)

(IK1 equals host 1´s facility interchange key)
(IK2 is used to exchange keys with host 2)

SYSTEM MEMORY AT HOST 1

i. E[IK1 XOR (i || i)](PWi)
j. E[IK1 XOR (j || j)](PWj)
k. E[IK1 XOR (k || k)](PWk)
.
.
.

USER i´s MEMORY AT HOST 1

Nonshared keys	Shared keys
E[IK1 XOR (i || i)](DK1)	E[IK2 XOR (i || j)](DKij)
E[IK1 XOR (i || i)](DK2)	(shared with user j at host 2)
E[IK1 XOR (i || i)](DK3)	
E[IK1 XOR (i || i)](DK4)	E[IK5 XOR (i || k)](DKik)
.	(shared with user k at host 5)
.	
.	E[IK7 XOR (i || m)](DKim)
.	(shared with user m at host 7)
.	.
.	.

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. SP 500-54	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

A KEY NOTARIZATION SYSTEM FOR COMPUTER NETWORKS

5. Publication Date

October 1979

6. Performing Organization Code

7. AUTHOR(S)

Miles E. Smid

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, DC 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Same as number 9

13. Type of Report & Period Covered

Final

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 79-600160

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

A cryptographic, key notarization system is proposed for computer networks to protect personal (nonshared) files, to communicate securely both on and off-line with local and remote users, to protect against key substitution, to authenticate system users, to authenticate data, and to provide a digital signature capability using a nonpublic key encryption algorithm. The system is implemented by addition of key notarization facilities which give users the capability of exercising a set of commands for key management as well as for data encryption functions. Key notarization facilities perform notarization which, upon encryption, seals a key or password with the identities of the transmitter and intended receiver.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Cryptography; digital signatures; encryption; identifiers; key management; key notarization

18. AVAILABILITY ☐ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-02130-0

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

19. SECURITY CLASS (THIS REPORT)	21. NO. OF PRINTED PAGES
UNCLASSIFIED	35
20. SECURITY CLASS (THIS PAGE)	**22. Price**
UNCLASSIFIED	$1.75

USCOMM-DC

☆U.S. GOVERNMENT PRINTING OFFICE: 1979-311-046/285

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $17; foreign $21.25. Single copy, $3 domestic; $3.75 foreign.

NOTE: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS—This monthly magazine is published to inform scientists, engineers, business and industry leaders, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing. Annual subscription: domestic $11; foreign $13.75.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide uscful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Services, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services, Springfield, VA 22161, in paper copy or microfiche form.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service. A literature survey issued biweekly. Annual subscription: domestic $25; foreign $30.

Liquefied Natural Gas. A literature survey issued quarterly. Annual subscription: $20.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $30. Please send subscription orders and remittances for the preceding bibliographic services to the National Bureau of Standards, Cryogenic Data Center (736) Boulder, CO 80303.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

SPECIAL FOURTH-CLASS RATE
BOOK